Curso Inteligência Artificial: do Zero ao Infinito

Redes Neurais Convolucionais

Universidade Federal de Mato Grosso

Agenda

1. Introdução

2. Convolução
   - Zero Padding
   - Stride
   - Dilatation
   - Formula
   - Pooling
   - Kernel
   - Channels
   - Number of Filters

Agenda

Convoluçao

- **Convoluçao** é a simples aplicação de um filtro sobre uma entrada resultando em uma ativação.
- A aplicação de forma repetida de um filtro (ou kernel) sobre uma imagem resulta em um *mapa de ativação*, também chamado **mapa de features**.
- A principal vantagem é que os filtros preservam as relações espaciais entre os pixels.

Convoluçao

- A inovaçao das **Redes Neurais Convolucionais** (CNNs) é a habilidade de automaticamente aprender um grande número de filtros.
- Camadas convolucionais podem conter diversos filtros, que são aplicados em paralelo.
- Como resultado, obtém-se diversos filtros especializados em extrair *features* específicas para determinadas tarefas.

Fonte: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

Agenda

1. Introdução

2. Convolução
   - Zero Padding
   - Stride
   - Dilatation
   - Formula
   - Pooling
   - Kernel
   - Channels
   - Number of Filters

Convolução

Convolução

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	?				

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	?				

$$\sum^{P} image_i \cdot K_i$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	?				

$$\sum^{P} image_i \cdot K_i = \boxed{5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1}$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	?				

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = \boxed{18}$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18				

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10			

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Revisão

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10	-3		

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10	-3	5	

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10	-3	5	
	12				

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Convolução

Image

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10	-3	5	
	12	?	?	?	
	?	?	?	?	
	?	?	?	?	

$$\sum^{P} image_i \cdot K_i = 5 \cdot 1 + 2 \cdot 0 + ... + 0 \cdot -1 = 18$$

Convolução

Image

	5	2	3	1	2	4
	2	4	1	0	3	1
	5	1	0	2	8	3
	0	2	1	5	2	4
	2	7	0	0	2	1
	1	3	2	8	7	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

	18	10	-3	5	
	12	?	?	?	
	?	?	?	?	
	?	?	?	?	

Convolução

Zero Padding

- A operação de convolução pode impactar na dimensão da imagem resultado, reduzindo-a consideravelmente
- **Padding** é o processo em que alguns pixels são adicionados ao redor da imagem
- Assim é possível manter a dimensionalidade na imagem resultante após uma operação de convolução

Fonte: Uma introdução as redes neurais convolucionais utilizando o Keras

Convolução

Zero Padding

Image

0	0	0	0	0	0	0	0
0	5	2	3	1	2	4	0
0	2	4	1	0	3	1	0
0	5	1	0	2	8	3	0
0	0	2	1	5	2	4	0
0	2	7	0	0	2	1	0
0	1	3	2	8	7	0	0
0	0	0	0	0	0	0	0

Kernel

1	0	0
2	1	0
1	0	-1

Output

-1					
	18	10	-3	5	
	12	?	?	?	
	?	?	?	?	
	?	?	?	?	

Convolução

Image

0	0	0	0	0	0	0	0
0	5	2	3	1	2	4	0
0	2	4	1	0	3	1	0
0	5	1	0	2	8	3	0
0	0	2	1	5	2	4	0
0	2	7	0	0	2	1	0
0	1	3	2	8	7	0	0
0	0	0	0	0	0	0	0

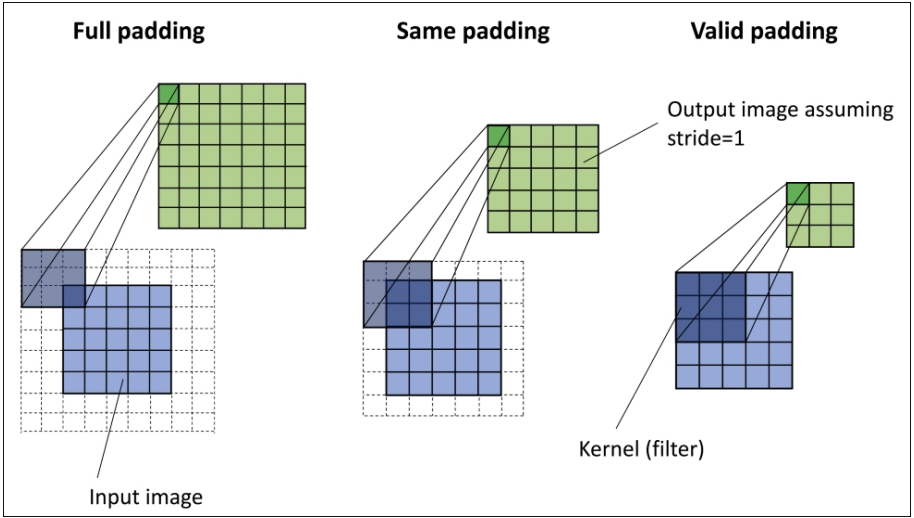
Kernel

1	0	0
2	1	0
1	0	-1

Output

-1	?	?	?	?	?
?	18	10	-3	5	?
?	12	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?

Convolução



Fonte: Classifying Images with Deep Convolutional Neural Networks

Convolução

Stride

- Algumas vezes é necessário "compactar" alguns pixels da imagem
- **Stride** indica o tamanho do deslocamento ao deslizar o *kernel* sobre a imagem
- Indicado em pixels

Convolução

Stride

Quando stride é 1, descola-se 1 pixel por vez

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

Stride

Quando stride é 1, descola-se 1 pixel por vez

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

Stride

Quando stride é 1, descola-se 1 pixel por vez

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

Stride

Quando stride é 2, descola-se 2 pixels por vez

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

Stride

Quando stride é 2, descola-se 2 pixels por vez

5	2	3	1	2	4
2	4	1	0	3	1
5	1	0	2	8	3
0	2	1	5	2	4
2	7	0	0	2	1
1	3	2	8	7	0

Convolução

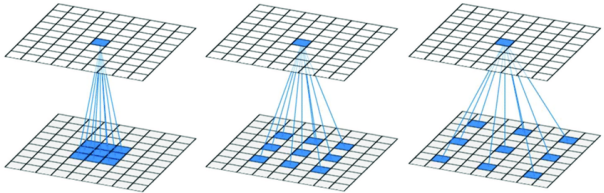
Dilatação

- Algumas vezes é interessante "pular" alguns valores. Ex. ao se trabalhar com imagens de grande dimensão
- **Convoluções Dilatadas** permitem que o *kernel* aumente o "alcance de visão", sem aumentar o número de parâmetros
- Também há perda de informação, pois nem todos os pixels são processados

Fonte: Types of Convolution Kernels : Simplified

Convolução

Dilatação



Fonte: Multiscale Spatial-Spectral Convolutional Network with Image-Based Framework for Hyperspectral Imagery Classification

Formula

A fórmula da convolução

$$Out = \frac{(In - F + 2P)}{S} + 1$$

em que

- In é o tamanho da entrada
- F é o tamanho do campo receptivo (kernel)
- P é a quantidade de *zero padding* na borda
- S é o *stride*

Referência: Convolutional Neural Networks for Visual Recognition

Pooling

- Outra estratégia para reduzir a dimensionalidade de uma imagem de entrada é utilizar camadas de **pooling**
- São também chamadas de camadas *downsampling* or *subsampling*
- Existem três diferentes tipos de *pooling* espacial:
  - Max Pooling
  - Average Pooling
  - Sum Pooling

Referência: Types of Convolution Kernels : Simplified

Max Pooling

Max Pooling: 2x2 stride 2

Input

-1	2	0	2
3	18	10	-3
2	12	5	2
1	3	7	4

Output

Max Pooling

Max Pooling: 2x2 stride 2

Input

-1	2	0	2
3	18	10	-3
2	12	5	2
1	3	7	4

Output

18	

Max Pooling

Max Pooling: 2x2 stride 2

Input

-1	2	0	2
3	18	10	-3
2	12	5	2
1	3	7	4

Output

18	10

Max Pooling

Max Pooling: 2x2 stride 2

Input

-1	2	0	2
3	18	10	-3
2	12	5	2
1	3	7	4

Output

10	18
12	

Max Pooling

Max Pooling: 2x2 stride 2

Input

-1	2	0	2
3	18	10	-3
2	12	5	2
1	3	7	4

Output

10	18
12	7

Kernel

- Detectores de *Features*.
- *Kernels* sao definidos durante o processo de treinamento.

Kernel

Blur

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

A simple blur done with convolutions

Fonte: Image convolution examples

Kernel

Gaussian Blur

0	0	0	5	0	0	0
0	5	18	32	18	5	0
0	18	64	100	64	18	0
5	32	100	100	100	32	5
0	18	64	100	64	18	0
0	5	18	32	18	5	0
0	0	0	5	0	0	0

Fonte: Image convolution examples

Kernel

Line Detection

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal lines

-1	2	-1
-1	2	-1
-1	2	-1

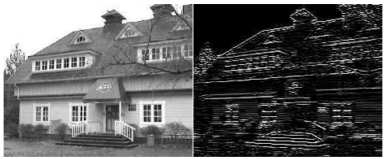
Vertical lines

-1	-1	2
-1	2	-1
2	-1	-1

45 degree lines

2	-1	-1
-1	2	-1
-1	-1	2

135 degree lines



Fonte: Image convolution examples

Kernel

Edge Detection

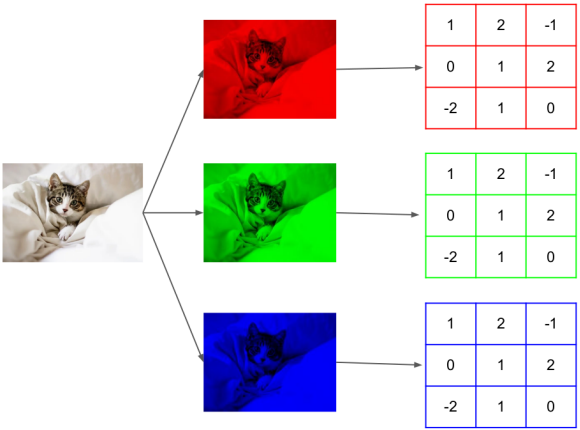
-1	-1	-1
-1	8	-1
-1	-1	-1

Fonte: Image convolution examples

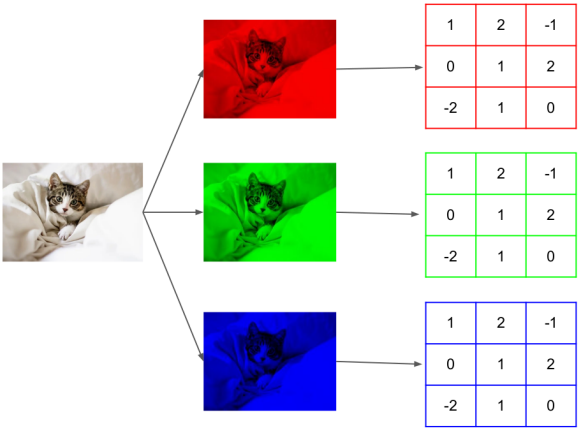
CNN Architecture

- Imagens coloridas possuem três canais: Red x Green x Blue

Channels

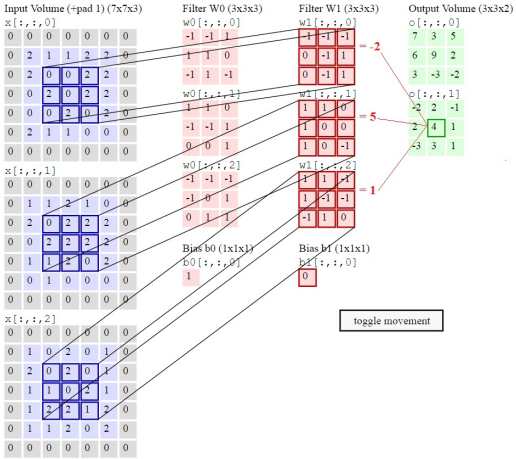


Channels



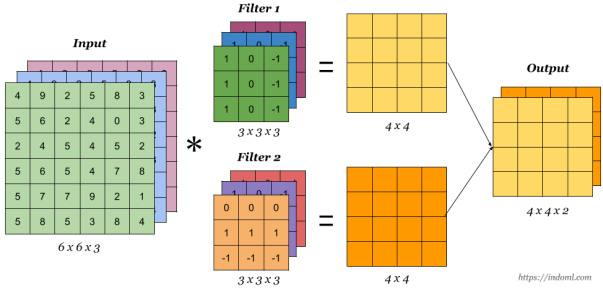
kernel = 3 x 3 x 3, Nº Pesos = 27

Number of Filters



Fonte: Convolutional Neural Networks (CNNs / ConvNets)

Number of Filters



Total de Pesos: $3x3x3 * 2 + 2 = 56$

Fonte: Implementing 'SAME' and 'VALID' padding of Tensorflow in Python

Referências

- Intuitively Understanding Convolutions for Deep Learning
  - https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1
- Convolutional Neural Networks, Explained
  - https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
- Image convolution examples
  - https://aishack.in/tutorials/image-convolution-examples/
- Types of Convolution Kernels : Simplified
  - https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37

Curso Inteligência Artificial: do Zero ao Infinito

Redes Neurais Convolucionais

Universidade Federal de Mato Grosso